

# PROJET : CHALLENGE PERSYCU

	NOM PRÉNOM	FONCTION	N° ÉTUDIANT
MAÎTRE D'OUVRAGE	PELLIER DAMIEN	ENSEIGNANT	—
MAÎTRES D'OEUVRE	ASOYAN ARMEN	ÉTUDIANT	11714531
	MACÉ GABRIEL	ÉTUDIANT	12010249
	VELLETAZ CÉDRIC	ÉTUDIANT	11908167
	ZARCH VASSILI	ÉTUDIANT	12107374

# Introduction

Le cahier présentant les différents tests fonctionnels et tests d'intégrations. Aucun test unitaire n'a été fait durant le projet. Il recouvre les différentes classes de java. La manière dont elles utilisent la librairie lejos et comment elles se vérifient dans la réalité.

## Tests fonctionnels

Description des scénarios (ainsi que leur enchaînement) permettant de vérifier que l'application recouvre bien le périmètre fonctionnel qui a été défini lors de la phase de spécification.

**Tableau 1: description des test fonctionnels**

Identification	Description	Contraintes	Dépendances	Procédure de test
Parcours linéaire	Mesurer la distance linéaire parcourue par le robot	Actionneur.java package motor: MovePilot.java	initialisation Motorisation	Une distance arbitraire  Mesurer la distance effectuée par le robot.
Rotation	Mesurer l'angle de rotation	Actionneur.java package motor: MovePilot.java	initialisation Motorisation	angle de rotation 360°  Le robot doit réaliser une rotation complète(angle 360°)  Mesurer l'erreur de l'angle de rotation donnée en paramètre et l'angle réel effectué.
Ajuster les Pince	Trouver la valeur optimale pour ouvrir et fermer les pinces.	lejos.hardware.motor.EV3MediumRegulatedMotor	initialisation pince	Le pilote contrôlant pince ouvre et ferme les pinces.  Mesure l'angle d'ouverture pour

				ajuster le paramètre de la méthode ouvrir.
Encodage de la valeur de distance.	Vérifier comment est encodée la valeur d'une distance dans le robot.	Le testeur positionne le robot perpendiculairement devant le mur.	initialisation Capteur ultrason	Affichage sur la console de la distance.  Mesure de la distance avec une règle.
Seuil linéaire de détection d'un palet	Trouver la distance minimal que le robot détecte un palet	Le testeur positionne le robot sur l'axe du capteur ultrason.  Il diminue la distance sur l'axe entre le robot et le palet	initialisation Capteur ultrason	Affichage sur la console de la distance.  Mesure de la distance avec une règle.  Arrêt de la procédure lorsque la distance affiche une valeur incohérente.
Seuil angulaire de détection d'un palet	Trouver l'angle maximal du seuil de détection du palet.	Le palet est positionné statiquement entre un mur et le robot.  La distance robot-mur sert de référence.	Encodage de la valeur de distance.  initialisation Capteur ultrason	Rotation du robot.  Arrêt de la procédure lorsque la distance affiche une valeur est la distance mur-robot.  Calcul de l'angle avec de la trigonométrie.
Initialisation position ligne de départ	L'utilisateur doit répondre à une question	L'utilisateur doit attendre l'affichage de la	le fichier agent.java package Motor	l'utilisateur appuie le bouton gauche, droite ou du

	affichée sur la console.	question sur la console	lejos.hardware.Button lejos.hardware.lcd.GraphicsLCD	<p>centre selon la position du robot sur la ligne de départ.</p> <p>L'attribut position à l'indice 0 contient la chaîne de caractère soit "Gauche", "Droite" ou "Milieu".</p> <p>La console affiche l'attribut position à l'indice 0 pour vérifier qu'elle est juste.</p>
Initialisation côté en-but	L'utilisateur doit répondre à une question affichée sur la console.	<p>La position du bas est définie par l'axe des abscisses de la caméra infrarouge.</p> <p>L'utilisateur doit attendre l'affichage de la question sur la console</p>	idem au dessus	<p>l'utilisateur appuie le bouton "Up" ou "Down" selon la position du robot sur la ligne de départ.</p> <p>L'attribut position à l'indice 1 contient la chaîne de caractère soit "Up" ou "Down".</p> <p>La console affiche l'attribut position à l'indice 1 pour vérifier qu'elle est juste.</p>
<b>premier palet</b>	Récupération du premier palet	L'utilisateur positionne le robot sur une des ligne de départ gauche ou droite	<p>Initialisation position ligne de départ</p> <p>le fichier agent.java package Motor</p>	<p>Ligne de départ gauche / droite</p> <p>itérations : jusqu'à trouver l'optimal : plus rapide et avec le moins d'erreur</p>

				Données mesurées à la main dans un premier temps!
<b>Second palet</b>	Récupération du second palet	Le robot est autonome sur le terrain	<b>premier palet</b>	idem au-dessus sauf donnée mesuré manuellement
<b>Troisième palet</b>	Récupération du troisième palet	idem au-dessus	<b>second palet</b>	idem au-dessus
<b>Chercher palet</b>	Recherche de palet	idem au-dessus	<b>Troisième palet</b> <b>Capteur US</b> <b>capteur touch</b>	test des mesures optimales à identifier comme un palet  Rotation jusqu'à trouver un palet : arrêt
<b>Orienter vers l'en-but</b>	Rechercher un angle pour s'orienter vers l'en-but de l'adversaire	idem au-dessus	<b>chercher palet</b>	les valeurs du capteur US.  trouver un angle pour s'orienter au mieux face à l'en but
<b>Avance vers l'en-but</b>	Atteindre l'en-but de l'adversaire	idem au-dessus	<b>Orienter vers l'en-but</b>	vérification: évitement obstacle sans perdre son orientation vers l'en-but, arrêt à la détection d'une ligne blanche

# Tests d'intégration

Description des tests (ainsi que leur enchaînement) permettant de vérifier que les différents paquets lejo et nos paquets s'interfaacent correctement. On distinguera les interfaces internes des interfaces externes à l'application.

**Tableau 2: Tests d'intégration**

Identification	Description	Contraintes	Dépendances	Procédure de test
Initialisation Motorisation	Vérifier si la classe se lance correctement	lejos.robotics.navigation.MovePilot  lejos.robotics.chassis.Wheel  lejos.hardware.motor.EV3LargeRegulatedMotor	Lire la console  Vérifier les branchements port-brick	donnée entrer: le offset définit la distance entre la roue centre du robot  Les deux ports EV3 duquel sont brancher les gros moteurs représenter par un chaine de caractère  Le diamètre de la roue  Si correcte ne renvoie aucune erreur  tester une méthode pour voir si le robot se déplace correctement
Initialisation Pinces	Vérifier si la classe se lance correctement	lejos.hardware.motor.EV3MediumRegulatedMotor	Lire la console  Vérifier les branchements port-brick	Le port EV3 de la pince représenter par un chaine de caractère  Si correcte ne renvoie aucune erreur  erreur de type Port sinon

Initialisation de touch	Vérifier si la classe se lance correctement	lejos.hardware.BrickFinder lejos.hardware.port.Port; lejos.hardware.sensor.EV3TouchSensor; lejos.hardware.Brick; lejos.robotics.SampleProvider;	Vérifier les branchements port-brick	Le port EV3 du capteur touch représenter par un chaîne de caractère  Si correcte ne renvoie aucune erreur
Initialisation capteur Ultrason	Vérifier si la classe se lance correctement	lejos.hardware.Brick lejos.hardware.BrickFinder lejos.hardware.port.Port; lejos.hardware.sensor.EV3UltrasonicSensor; lejos.robotics.SampleProvider;  java.util.Arrays;	Vérifier les branchements port-brick	Le port EV3 de capteur US représenter par un chaîne de caractère  Si correcte ne renvoie aucune erreur  erreur Port sinon  tester la méthode renvoyant la valeur de la distance UV
Initialisation Infrarouge	Vérifier si la classe se lance correctement	voir import de CameraInfrarouge.java	Vérifier les branchements port-brick	Le port EV3 de capteur IR représenter par un chaîne de caractère  num port adresse IP  Si correcte ne renvoie aucune erreur  sinon affichage d'une erreur de type SocketException ou UnknownHostException sur la console

## Références

GitHub - InitiationIA/InitiationIA. (s. d.). GitHub. <https://github.com/InitiationIA/InitiationIA>

leJOS. (s. d.). <https://sourceforge.net/p/lejos/wiki/Home/>